Fast High-Accuracy Multi-Dimensional
Pattern Inspection

Field of the Invention

This invention relates to machine vision, and particularly to systems for pattern inspection in an

image.

1    Background of the Invention

2    Digital images are formed by many devices and used for many practical purposes. Devices

3    include TV cameras operating on visible or infrared light, line-scan sensors, flying spot scanners,

4    electron microscopes, X-ray devices including CT scanners, magnetic resonance imagers, and

5    other devices known to those skilled in the art. Practical applications are found in industrial

6    automation, medical diagnosis, satellite imaging for a variety of military, civilian, and scientific

7    purposes, photographic processing, surveillance and traffic monitoring, document processing,

8    and many others.

9    To serve these applications the images formed by the various devices are analyzed by digital

10   devices to extract appropriate information. One form of analysis that is of considerable practical

11   importance is determining the position, orientation, and size of patterns in an image that

12   correspond to objects in the field of view of the imaging device. Pattern location methods are of

1 particular importance in industrial automation, where they are used to guide robots and other

2 automation equipment in semiconductor manufacturing, electronics assembly, pharmaceuticals,

3 food processing, consumer goods manufacturing, and many others.

4 Another form of digital image analysis of practical importance is identifying differences between

5 an image of an object and a stored pattern that represents the "ideal" appearance of the object.

6 Methods for identifying these differences are generally referred to as pattern inspection methods,

7 and are used in industrial automation for assembly, packaging, quality control, and many other

8 purposes.

9 One early, widely-used method for pattern location and inspection is known as blob analysis. In

10 this method, the pixels of a digital image are classified as "object" or "background" by some

11 means, typically by comparing pixel gray-levels to a threshold. Pixels classified as object are

12 grouped into blobs using the rule that two object pixels are part of the same blob if they are

13 neighbors; this is known as connectivity analysis. For each such blob we determine properties

14 such as area, perimeter, center of mass, principal moments of inertia, and principal axes of

15 inertia. The position, orientation, and size of a blob is taken to be its center of mass, angle of first

16 principal axis of inertia, and area, respectively. These and the other blob properties can be

17 compared against a known ideal for proposes of inspection.

18 Blob analysis is relatively inexpensive to compute, allowing for fast operation on inexpensive

19 hardware. It is reasonably accurate under ideal conditions, and well-suited to objects whose

20 orientation and size are subject to change. One limitation is that accuracy can be severely

21 degraded if some of the object is missing or occluded, or if unexpected extra features are present.

22 Another limitation is that the values available for inspection purposes represent coarse features of

1 the object, and cannot be used to detect fine variations. The most severe limitation, however, is

2 that except under limited and well-controlled conditions there is in general no reliable method for

3 classifying pixels as object or background. These limitations forced developers to seek other

4 methods for pattern location and inspection.

5 Another method that achieved early widespread use is binary template matching. In this method,

6 a training image is used that contains an example of the pattern to be located. The subset of the

7 training image containing the example is thresholded to produce a binary pattern and then stored

8 in a memory. At run-time, images are presented that contain the object to be found. The stored

9 pattern is compared with like-sized subsets of the run-time image at all or selected positions, and

10 the position that best matches the stored pattern is considered the position of the object. Degree

11 of match at a given position of the pattern is simply the fraction of pattern pixels that match their

12 corresponding image pixel, thereby providing pattern inspection information.

13 Binary template matching does not depend on classifying image pixels as object or background,

14 and so it can be applied to a much wider variety of problems than blob analysis. It also is much

15 better able to tolerate missing or extra pattern features without severe loss of accuracy, and it is

16 able to detect finer differences between the pattern and the object. One limitation, however, is

17 that a binarization threshold is needed, which can be difficult to choose reliably in practice,

18 particularly under conditions of poor signal-to-noise ratio or when illumination intensity or

19 object contrast is subject to variation. Accuracy is typically limited to about one whole pixel due

20 to the substantial loss of information associated with thresholding. Even more serious, however,

21 is that binary template matching cannot measure object orientation and size. Furthermore,

1 accuracy degrades rapidly with small variations in orientation and/or size, and if larger variations

2 are expected the method cannot be used at all.

3 A significant improvement over binary template matching came with the advent of relatively

4 inexpensive methods for the use of gray-level normalized correlation for pattern location and

5 inspection. The methods are similar, except that no threshold is used so that the full range of

6 image gray-levels are considered, and the degree of match becomes the correlation coefficient

7 between the stored pattern and the image subset at a given position.

8 Since no binarization threshold is needed, and given the fundamental noise immunity of

9 correlation, performance is not significantly compromised under conditions of poor signal-to-

10 noise ratio or when illumination intensity or object contrast is subject to variation. Furthermore,

11 since there is no loss of information due to thresholding, position accuracy down to about ¼ pixel

12 is practical using well-known interpolation methods. The situation regarding orientation and size,

13 however, is not much improved with respect to binary template matching. Another limitation is

14 that in some applications, contrast can vary locally across an image of an object, resulting in poor

15 correlation with the stored pattern, and consequent failure to correctly locate it.

16 More recently, improvements to gray-level correlation have been developed that allow it to be

17 used in applications where significant variation in orientation and/or size is expected. In these

18 methods, the stored pattern is rotated and/or scaled by digital image re-sampling methods before

19 being matched against the image. By matching over a range of angles, sizes, and x-y positions,

20 one can locate an object in the corresponding multidimensional space. Note that such methods

21 would not work well with binary template matching, due the much more severe pixel

22 quantization errors associated with binary images.

1 One problem with these methods is the severe computational cost, both of digital re-sampling

2 and of searching a space with more than 2 dimensions. To manage this cost, the search methods

3 break up the problem into two or more phases. The earliest phase uses a coarse, subsampled

4 version of the pattern to cover the entire search space quickly and identify possible object

5 locations in the n-dimensional space. Subsequent phases use finer versions of the pattern to

6 refine the locations determined at earlier phases, and eliminate locations that the finer resolution

7 reveals are not well correlated with the pattern. Note that variations of these coarse-fine methods

8 have also been used with binary template matching and the original 2-dimensional correlation,

9 but are even more important with the higher-dimensional search space.

10 The location accuracy of these methods is limited both by how finely the multidimensional space

11 is searched, and by the ability of the discrete pixel grid to represent small changes in position,

12 orientation, and scale. The fineness of the search can be chosen to suit a given application, but

13 computational cost grows so rapidly with resolution and number of dimensions that practical

14 applications often cannot tolerate the cost or time needed to achieve high accuracy. The

15 limitations of the discrete pixel grid are more fundamental — no matter how finely the space is

16 searched, for typical patterns one cannot expect position accuracy to be much better than about ¼

17 pixel, orientation better than a degree or so, and scale better than a percent or so.

18 A similar situation holds when gray-level pixel-grid-based methods are used for pattern

19 inspection. Once the object has been located in the multidimensional space, pixels in the pattern

20 can be compared to each corresponding pixel in the image to identify differences. Some

21 differences, however, will result from the re-sampling process itself, because again the pixel grid

22 cannot accurately represent small variations in orientation and scale. These differences are

1 particularly severe in regions where image gray levels are changing rapidly, such as along object

2 boundaries. Often these are the most important regions of an object to inspect. Since in general,

3 differences due to re-sampling cannot be distinguished from those due to object defects,

4 inspection performance is compromised.

5

## Summary of the Invention

7 In one general aspect, the invention is a method and apparatus for identifying differences

8 between a stored pattern and a matching image subset, where variations in pattern position,

9 orientation, and size do not give rise to false differences. The process of identifying differences is

10 called *inspection*. Generally, an object image must be precisely located prior to inspection. In

11 another general aspect, the invention is a system for analyzing an object image with respect to a

12 model pattern, wherein the system includes extracting pattern features from the model pattern;

13 generating a vector-valued function using the pattern features to provide a pattern field;

14 extracting image features from the object image; evaluating each image feature, using the pattern

15 field and an n-dimensional transformation that associates image features with pattern features, so

16 as to determine at least one associated feature characteristic; and using at least one feature

17 characteristic to identify at least one flaw in the object image. In a preferred embodiment, at least

18 one associated feature characteristic includes a probability value that indicates the likelihood that

19 an associated image feature does not correspond to a feature in the model pattern. In an alternate

20 preferred embodiment, the at least one associated feature characteristic includes a probability

21 value that indicates the likelihood that an associated image feature does correspond to a feature in

22 the model pattern.

1    In another preferred embodiment, at least one pattern feature includes a probability value

2    indicating the likelihood that the pattern feature does not correspond to at least one feature in the

3    object image.

4    When using at least one feature characteristic, it is preferred to transfer a feature

5    characteristic from the at least one image feature to an element of the pattern field, where in a

6    preferred embodiment, the element of the pattern field is the nearest element of the pattern field.

7    When using at least one feature characteristic, it is also preferred to use a plurality of

8    image features; and transfer a plurality of the feature characteristics from the plurality of image

9    features to a plurality of elements of the pattern field, wherein some of the plurality of elements

10    of the pattern field can include at least one link to a neighboring element of the pattern field.

11    Further, after transferring a plurality of the feature characteristics from the plurality of image

12    features to a plurality of elements of the pattern field, it is preferred that each element of the

13    plurality of elements of the pattern field receive a feature characteristic equal to the maximum of

14    its own feature characteristic and the feature characteristic of each neighboring element of the

15    pattern field.

16    In another preferred embodiment, to use at least one feature characteristic includes

17    identifying the nearest element of the pattern field; transferring a feature characteristic from the

18    at least one image feature to the nearest element of the pattern field; and computing a coverage

19    value using at least the transferred feature characteristic.

20    In a further preferred embodiment, evaluating each image feature includes comparing the

21    direction of each image feature with the direction of an element of the pattern field. It is

22    preferable to assign a higher weight to the image feature if the difference in the direction of the

23    image feature from the direction of an element of the pattern field is less than a specified

1 direction parameter. The specified direction parameter can be determined by a characteristic of

2 the element of the pattern field, such as a flag indicating "corner" or "non-corner".

3 It is also possible for a lower weight to be assigned to the image feature if the difference

4 in the direction of the image feature from the direction of an element of the pattern field is greater

5 than a specified direction parameter, wherein the specified direction parameter can be determined

6 by a characteristic of the element of the pattern field, such as a flag indicating "corner" or "non-

7 corner".

8 In yet another preferred embodiment, to evaluate each image feature includes comparing,

9 modulo 180 degrees, the direction of each image feature with the direction of an element of the

10 pattern field.

11 It is also possible that to evaluate each image feature includes assigning a weight of zero

12 when the image feature is at a position that corresponds to an element of the pattern field that

13 specified that no image feature is expected at that position.

14 Moreover, to evaluate each image feature can include comparing the distance of each

15 image feature with a specified distance parameter, where a lower weight can be assigned to the

16 image feature if the distance of the image feature is greater than a specified distance parameter,

17 or alternatively, where a higher weight can be assigned to the image feature if the distance of the

18 image feature is less than a specified distance parameter.

19 In another preferred embodiment, to evaluate each image feature includes comparing the

20 direction of each image feature with the direction of an element of the pattern field, and

21 comparing the distance of each image feature with a specified distance parameter.

22 To avoid ambiguity we will call the location of a pattern in a multidimensional space its

23 *pose*. More precisely, a pose is a coordinate transform that maps points in an image to

1 corresponding points in a stored pattern. In a preferred embodiment, a pose is a general six

2 degree of freedom linear coordinate transform. The six degrees of freedom can be represented by

3 the four elements of a 2x2 matrix, plus the two elements of a vector corresponding to the two

4 translational degrees of freedom. Alternatively and equivalently, the four non-translational

5 degrees of freedom can be represented in other ways, such as orientation, scale, aspect ratio, and

6 skew, or x-scale, y-scale, x-axis-angle, and y-axis-angle.

7 The invention can serve as a replacement for the fine resolution phase of any coarse-fine

8 method for pattern inspection, such as the prior art method of correlation search followed by

9 Golden Template Analysis. In combination with the coarse location phases of any such method,

10 the invention results in an overall method for pattern inspection that is faster and more accurate

11 than any known prior art method.

12 In a preferred embodiment, the PatQuick™ tool, sold by Cognex Corporation, Natick

13 MA, is used for producing an approximate object pose.

14 The invention uses a stored pattern that represents an ideal example of the object to be

15 found and inspected. The pattern can be created from a training image or synthesized from a

16 geometric description. According to the invention, patterns and images are represented by a

17 feature-based description that can be translated, rotated, and scaled to arbitrary precision much

18 faster than digital image re-sampling, and without pixel grid quantization errors. Thus accuracy is

19 not limited by the ability of a grid to represent small changes in position, orientation, or size (or

20 other degrees of freedom). Furthermore, pixel quantization errors due to digital re-sampling will

21 not cause false differences between the pattern and image that can limit inspection performance,

22 since no re-sampling is done.

1    Accuracy is also not limited by the fineness with which the space is searched, because the

2    invention does not test discrete positions within the space to determine the pose with the highest

3    degree of match. Instead the invention determines an accurate object pose from an approximate

4    starting pose in a small, fixed number of increments that is independent of the number of

5    dimensions of the space (i.e. degrees of freedom) and independent of the distance between the

6    starting and final poses, as long as the starting pose is within some "capture range" of the true

7    pose. Thus one does not need to sacrifice accuracy in order to keep execution time within the

8    bounds allowed by practical applications.

9    Unlike prior art methods where execution time grows rapidly with number of degrees of

10   freedom, with the method of the invention execution time grows at worst very slowly, and in

11   some embodiments not at all. Thus one need not sacrifice degree of freedom measurements in

12   order to keep execution time within practical bounds. Furthermore, allowing four or more

13   degrees of freedom to be refined will often result in better matches between the pattern and

14   image, and thereby improved accuracy.

15   The invention processes images with a feature detector to generate a description that is

16   not tied to a pixel grid. The description is a list of elements called *dipoles* (also called features)

17   that represent points (positions) along object boundaries. A dipole includes the coordinates of the   .

18   position of a point along an object boundary and a direction pointing substantially normal to the

19   boundary at that point. Object boundaries are defined as places where image gradient (a vector

20   describing rate and direction of gray-level change at each point in an image) reaches a local

21   maximum. In a preferred embodiment, gradient is estimated at an adjustable spatial resolution. In

22   another preferred embodiment, the dipole direction is the gradient direction. In another preferred

1    embodiment, a dipole, i.e., a feature, contains additional information as further described in the

2    drawings. In yet another preferred embodiment, dipoles are generated not from an image but

3    from a geometric description of an object, such as might be found in a CAD system.

4    The stored model pattern to be used by the invention for localization and subsequent

5    inspection is the basis for generating a dipole list that describes the objects to be found by

6    representing object boundaries. The dipole list derived from the model pattern is called the *field*

7    *dipole list*. It can be generated from a model training image containing an example object using a

8    feature detector, or it can be synthesized from a geometric description. The field dipole list is

9    used to generate a 2-dimensional vector-valued function called a *field*. For each point within the

10   region of the stored model pattern , the field gives a vector that indicates the distance and

11   direction to the nearest point along a model object boundary. The vector is called the *force* at the

12   specified point within the stored model pattern.

13   Note that the nearest point along a model object boundary is not necessarily one of the

14   model object boundary points represented by the field dipoles, but in general may lie between

15   field dipole positions. Note further that the point within the stored model pattern is not

16   necessarily an integer grid position, but is in general a real-valued position, known to within the

17   limits of precision of the apparatus used to perform the calculations. Note that since the force

18   vector points to the nearest boundary point, it must be normal to the boundary (except at

19   discontinuities).

20   In a preferred embodiment, if no model object boundary point lies within a certain range

21   of a field position, then a special code is given instead of a force vector. In another preferred

22   embodiment, the identity of the nearest field dipole is given in addition to the force. In another

- 11 -

1    preferred embodiment, one additional bit of information is given that indicates whether the

2    gradient direction at the boundary pointed to by the force is the same or $180°$ opposite from the

3    force direction (both are normal to the boundary). In another preferred embodiment, additional

4    information is given as further described in the drawings. In another embodiment, the field takes

5    a direction in addition to a position within the pattern, and the force returned is the distance and

6    direction to the nearest model object boundary point in approximately the given direction.

7         The stored model pattern used by the invention includes the field dipole list, the field, and

8    a set of operating parameters as appropriate to a given embodiment, and further described

9    throughout the specification.

10        Given an object image and an approximate starting pose, pattern localization proceeds as

11   follows. The object image is processed by a feature detector to produce a dipole list, called the

12   *image dipole list*. The starting pose is refined in a sequence of incremental improvements called

13   *attraction steps*. Each such step results in a significantly more accurate pose in all of the degrees

14   of freedom that are allowed to vary. The sequence can be terminated after a fixed number of

15   steps, and/or when no significant change in pose results from the last step, or based on any

16   reasonable criteria. In a preferred embodiment, the sequence is terminated after four steps.

17        For each attraction step, the image dipoles are processed in any convenient order. The

18   position and direction of each image dipole is mapped by the current pose transformation to

19   convert image coordinates to model pattern (field) coordinates. The field is used to determine the

20   force at the point to which the image dipole was mapped. Since each image dipole is presumed to

21   be located on an object boundary, and the force gives the distance and direction to the nearest

22   model object boundary of the stored model pattern, the existence of the image dipole at the

1    mapped position is taken as evidence that the pose should be modified so that the image dipole

2    moves in the force direction by an amount equal to the force distance.

3        It is important to note that object boundaries generally provide position information in a

4    direction normal to the boundary, which as noted above is the force direction, but no information

5    in a direction along the boundary. Thus the evidence provided by an image dipole constrains a

6    single degree of freedom only, specifically position along the line of force, and provides no

7    evidence in the direction normal to the force.

8        If the current pose is a fair approximation to the true object position, then many image

9    dipoles will provide good evidence as to how the pose should be modified to bring the image

10   boundaries into maximum agreement with the boundaries of the stored model pattern. For a

11   variety of reasons, however, many other image dipoles may provide false or misleading evidence.

12   Thus, it is important to evaluate the evidence provided by each image dipole, and assign a

13   weighting factor to each image dipole to indicate the relative reliability of the evidence.

14       In one embodiment, the direction (as mapped to the pattern coordinate system) of an

15   image dipole is compared with the force direction, and the result, modulo 180°, is used to

16   determine the weight of the image dipole. If the directions agree to within some specified

17   parameter, the dipole is given a high weight; if they disagree beyond some other specified

18   parameter, the dipole is given zero weight; if the direction difference falls between the two

19   parameters, intermediate weights are assigned.

20       In another embodiment, the image dipole direction is compared to the gradient direction

21   of the model pattern boundary to which the force points. A parameter is used to choose between

1 making the comparison modulo 180°, in which case gradient polarity is effectively ignored, or

2 making it modulo 360°, in which case gradient polarity is considered. In a preferred embodiment,

3 the field itself indicates at each point within the stored model pattern whether to ignore polarity,

4 consider polarity, or defer the decision to a global parameter.

5 In one embodiment, the force distance is used to determine the dipole weight. In a

6 preferred embodiment, if the force distance is larger than some specified parameter, the dipole is

7 given zero weight, on the assumption that the dipole is too far away to represent valid evidence.

8 If the force distance is smaller than some other specified parameter, the dipole is given a high

9 weight, and if it falls between the two parameters, intermediate weights are assigned.

10 In a preferred embodiment, the parameters specifying the weight factor as a function of

11 force distance are adjusted for each attraction step to account for the fact that the pose is

12 becoming more accurate, and therefore that one should expect image dipoles representing valid

13 evidence to be closer to the pattern boundaries.

14 In one embodiment, the gradient magnitude of the image dipoles is used to determine the

15 dipole weight. In a preferred embodiment, a combination of dipole direction, force distance, and

16 gradient magnitude is used to determine the weight.

17 For each attraction step, the invention determines a new pose that best accounts for the

18 evidence contributed by each image dipole, and taking into account the dipole's weight. In a

19 preferred embodiment, a least-squares method is used to determine the new pose.

20 The evaluation of each image dipole to produce a weight can also provide information for

21 inspection purposes. It is desirable to look for two distinct kinds of errors: missing features,

1　which are pattern features for which no corresponding image feature can be found, and extra

2　features, sometimes called "clutter", which are image features that correspond to no pattern

3　feature. In one embodiment, image dipoles with low weights are considered to be clutter. In a

4　preferred embodiment, a specific clutter value is computed for each image dipole, as further

5　described in the drawings below.

6　.　In an embodiment of the invention that can identify missing pattern features, the field at

7　each point gives identity of the nearest field dipole, if any, in addition to the force vector. Each

8　field dipole contains an evaluation, which is initialized to zero. Each image dipole transfers its

9　evaluation (also called "weight" or "feature characteristic") to that of the nearest field dipole as

10　indicated by the field. Since in general the correspondence between image and field dipoles is not

11　one-to-one, some field dipoles may receive evaluations (feature characteristics) from more than

12　one image dipole, and others may receive evaluations from none. Those field dipoles that receive

13　no evaluation may represent truly missing features, or may simply represent gaps in the transfer

14　due to quantization effects.

15　When more than one evaluation is transferred to a given field dipole, the evaluations can

16　be combined by any reasonable means. In a preferred embodiment, the largest such evaluation is

17　used and the others are discarded. Gaps in the transfer can be closed by considering neighboring

18　field dipoles. In one embodiment, methods known in the art as gray-level mathematical

19　morphology are used to close the gaps. In the case of the invention, one-dimensional versions of

20　morphological operations are used, since field dipoles lie along one-dimensional boundaries. In a

21　preferred embodiment, a morphological dilation operation is used.

1   If the starting pose is too far away from the true pose, there may be insufficient good evidence

2   from the image dipoles to move the pose in the right direction. The set of starting poses that

3   result in attraction to the true pose defines the *capture range* of the pattern. The capture range

4   depends on the specific pattern in use, and determines the accuracy needed from whatever

5   method is used to determine the starting pose.

6   In a preferred embodiment, the feature detector that is used to generate dipoles is tunable over a

7   wide range of spatial frequencies. If the feature detector is set to detect very fine features at a

8   relatively high resolution, the accuracy will be high but the capture range will be relatively small.

9   If on the other hand the feature detector is set to detect coarse features at a relatively lower

10  resolution, the accuracy will be lower but the capture range will be relatively large. This suggests

11  a multi-resolution method where a coarse, low resolution step is followed by a fine, high

12  resolution step. With this method, the capture range is determined by the coarse step and is

13  relatively large, while the accuracy is determined by the fine step and is high.

14

15  **Brief Description of the Drawing**

16  The invention will be more fully understood from the following detailed description, in

17  conjunction with the following figures, wherein:

18      Fig. 1 is a high-level block diagram of an embodiment of the invention;

19      Fig. 1A is an illustration of a pixel array having a 2:1 aspect ratio;

1 Fig. 2 is a block diagram of the training block of Fig. 1;

2 Fig. 3 is a block diagram of the feature detection block of Fig. 2;

3 Fig. 4 is a diagram showing bit assignments of a 32-bit word, and optional 'nearest

4 dipole' bits;

5 Fig. 5 is an illustration of a field element array, including a border of field elements;

6 Fig. 6 illustrates field seeding, showing some of the field elements of Fig. 5, including a

7 plurality of straight line segments of a pattern boundary, and the associated field dipoles;

8 Fig. 7A illustrates field dipole connecting, showing some of the field elements of Fig. 6,

9 including a plurality of straight line segments of a pattern boundary, and a plurality of associated

10 right and left links;

11 Figs. 7B and 7C are diagrams illustrating the order in which neighboring field elements

12 are examined;

13 Fig. 8 illustrates chain segmentation of Fig. 2, showing some of the field elements of Fig.

14 5, including a plurality of straight line segments, and a plurality of left and right links;

15 Figs. 9A, 9B, and 9C illustrate part of the analysis that is performed by the propagate

16 phase of the field generation module of Fig. 2;

17 Fig. 10 shows further details of the propagate phase of the field generation module of Fig.

18 2;

1    Fig. 11 shows the same portion of the field array that was shown after seeding in Fig. 6,

2    but with new force vectors resulting from one propagation;

3    Fig. 12 shows the same portion of the field array as Fig. 11, after two propagations;

4    Fig. 13 is a block diagram of the run-time module of the preferred embodiment of Fig. 1;

5    Fig. 14 is a diagram illustrating a least squares method for determining a pose that best

6    accounts for the evidence of the image dipoles at each attraction step;

7    Fig. 15 is a block diagram of the attraction module of Fig. 13;

8    Fig. 16 is a block diagram of the map module of Fig. 15;

9    Fig. 17 is a block diagram of the field module of Fig. 15;

10    Fig. 18 is a block diagram of the rotate module of Fig. 15;

11    Figs 19A, 19B, and 19C show output as a function of input for three fuzzy logic

12    processing elements;

13    Fig. 20A is a schematic diagram of the portion of the evaluate module of Fig. 15, showing

14    a preferred system for calculating 'weight' and 'eval';

15    Fig. 20B is a schematic diagram of the portion of the evaluate module of Fig. 15, showing

16    a preferred system for calculating 'clutter';

17    Fig. 21 is a schematic diagram of the sum module of Fig. 15;

1       Figs. 22A-D are block diagram of the solve module of Fig. 15, showing the equations and

2    inputs for providing 'motion' and 'rms error' for various degrees of freedom ;

3       Fig. 23 is a block diagram of the equations of the compose module of Fig. 15;

4       Fig. 24 is a block diagram of the equations of the Normal Tensor module of Fig. 15;

5       Fig. 25 is a graphical illustration of a plurality of image dipoles and a plurality of

6    connected field dipoles, showing field dipole evaluation;

7       Fig. 26 is a high-level block diagram of a multi-resolution embodiment of the invention;

8       Figs. 27A and 27B are flow diagrams illustrating the sequence of operations performed

9    by the modules of Fig. 15; and

10       Fig. 28 is a flow diagram illustrating a multi-resolution mode of operation of the

11    invention.

12

13

14    **Detailed Description of the Preferred Embodiments**

15    In the following figures, "modules" can be implemented as software, firmware, or hardware.

16    Moreover, each module may include sub-modules, or "steps", each of which can be implemented

17    as either hardware, software, or some combination thereof. Figure 1 is a high-level block

18    diagram of one embodiment of the invention. A training (model) image **100** containing an

19    example of a pattern **105** to be used for localization and/or inspection is presented. A training

1     module 110 analyzes the training image and produces a stored model pattern 120 for subsequent

2     use. At least one run-time image 130 is presented, each such image containing zero or more

3     instances of patterns 135 similar in shape, but possibly different in size and orientation, to the

4     training (model) pattern 105.

5     Each run-time image 130 has an associated *client map 131*, chosen by a user for a particular

6     application. A client map is a coordinate transformation that maps, i.e., associates points in an

7     orthonormal but otherwise arbitrary coordinate system to points in the image. An orthonormal

8     coordinate system has perpendicular axes, each axis having a unit scale. The client map provides

9     an orthonormal reference that is necessary to properly handle the orientation degree of freedom,

10     as well as the skew, scale, and aspect ratio degrees of freedom . In practical applications, the

11     images themselves are almost never orthonormal, since practical image sensors almost never

12     have perfectly square pixels. For example, pixels having an aspect ration of 2:1 are possible, as

13     shown in Fig. 1A. In this case, the client map would be a 2x2 matrix:

14

15             0.5   0.0

16             0.0   1.0

17

18     If the pixels are square, the client map is the identity transform, i.e., each diagonal entry in the

19     transform matrix is 1.0, and each off-diagonal element is 0.0.

1    Furthermore, it is sometimes useful to have a significantly non-orthonormal field. For example, a

2    field generated from a square pattern can be used to localize and inspect a rectangular or even

3    parallelogram-shaped instance of the pattern by using an appropriate starting pose. These cases

4    can only be handled if an orthonormal reference is available.


5    For each run-time image, a starting pose 132 is determined by any suitable method, such as

6    coarse gray-level correlation with orientation and size re-sampling, a coarse generalized Hough

7    transform, or the Cognex PatQuick™ tool. The starting pose 132 is a six-degree-of-freedom

8    coordinate transformation that maps points in the pattern 105 to approximately corresponding

9    points 135 in the run-time image 130. A run-time module 140 analyzes the image 130, using the

10   stored pattern 120, the starting pose 132, and the client map 131. As a result of the analysis, the

11   run-time module 140 produces a pose 134 that maps pattern points 105 to accurately

12   corresponding image points 135.


13   The run-time module 140 produces an *rms error* value 136 that is a measure of the degree of

14   match between the pattern 105 and the image 130. The rms error value 136 is the root-mean-

15   square error from the least squares solution, or other error minimization solution, that is used to

16   determine a pose that best fits the evidence of the image dipoles, to be described in more detail

17   below. A value of zero represents a perfect fit, while higher values represent poorer fits.


18   The run-time module 140 produces a *coverage* value 138 that is a measure of the fraction of the

19   pattern 105 to which corresponding image features have been found. The coverage value 138 is

20   computed by summing the field dipole evaluations and dividing by the number of field dipoles,

21   to be described in more detail below.

1 The run-time module 140 produces a *clutter* value that is a measure of extra features found in the

2 image that do not correspond to pattern features. In a preferred embodiment, the clutter value is

3 computed by summing the individual image dipole clutter values and dividing by the number of

4 field dipoles.

5 The run-time module 140 produces an evaluated image dipole list 150 and an evaluated field

6 dipole list 160. The evaluated image dipole list 150 identifies features in the image 130 not

7 present in the pattern 105, and the evaluated field dipole list 160 identifies features in the pattern

8 105 not present in the image 130. The differences between the image and pattern so identified

9 can be used for inspection purposes.

10 Figure 2 shows a block diagram of the training module 110 and pattern-related data 120. A

11 training image 100 containing an example of a pattern 105 to be localized and/or inspected is

12 presented to training module 110, which analyzes the training image 100 and produces a stored

13 pattern 120 for subsequent use. The training module 110 consists of two modules, a feature

14 detection module 200 and a field generation module 210. Both modules use various parameters

15 220 to control operation as appropriate for the application. These parameters, as well as those

16 needed for subsequent run-time modules, are pattern-dependent and therefore are collected and

17 stored in the pattern 120 as shown.

18 The feature detection module 200 processes the training image 100, and using the parameters

19 220, to produce a field dipole set 230, which is stored in the pattern 120 as shown. The field

20 generation module 210 uses the field dipole set 230 and parameters 220 to produce a field 240,

21 stored in the pattern 120 as shown.

1   As described in the summary above, the field 240 produces information as a function of

2   theoretically real-valued position within the region of the pattern 105. In practice, since all the

3   field values cannot be computed analytically and stored, a 2-dimensional array is used that stores

4   field values at discrete points on a grid. Given a real-valued position (to some precision

5   determined by the particular details of the embodiment), an interpolation method, such as the

6   method shown in Fig. 17, is used to compute field values at intermediate points between grid

7   elements. Since the field grid is never translated, rotated, or scaled, no re-sampling is needed and

8   grid quantization effects are small. Instead, the image dipoles, which are not grid-based, are

9   mapped to the fixed field coordinates in accordance with the map of Fig. 16.

10  Thus the purpose of the field generation module **210** is to compute the elements of the 2-

11  dimensional array that encodes the field **240**. The field generation module 210 is itself composed

12  of many steps or submodules, as shown in figure 2. Each of these steps modifies the field in

13  some way, generally based on the results of the previous steps. Some of the steps also add

14  information to the field dipole set **230**. The specific sequence of steps shown in figure 2

15  corresponds to a preferred embodiment; many other variations are possible within the spirit of

16  the invention; the essential requirement is that the stored pattern 120 be able to provide certain

17  information as a function of position within the region of the pattern 105.

18  In a preferred embodiment as shown in figure 2, the field generation module 210 consists of the

19  following steps. An initialization step **250** loads predefined codes into the field array elements

20  520, 540, shown in Fig. 5. A seed step **252** sets up field array elements at positions corresponding

21  to the dipoles in the field dipole set **230**. A connect step **254** uses the seeded field array to

22  identify neighboring dipoles for each field dipole. Once identified, the field dipoles are connected

1    to neighboring ones to form chains by storing the identity of left and right neighbors along

2    pattern boundaries, if any. A chain step 256 scans the connected field dipoles to identify and

3    catalog discrete chains. For each chain, the starting and ending dipoles, length, total gradient

4    magnitude, and whether the chain is open or closed is determined and stored.

5    A filter step 258 removes weak chains from the pattern by removing the dipoles they contain

6    - from the field array (i.e. reversing the seeding step 252 for those dipoles). A variety of criteria

7    can be used to identify weak chains. In a preferred embodiment, chains whose total gradient

8    magnitude or average gradient magnitude are below some specified parameter are considered

9    weak.

10    A segment step 260 divides chains into segments of low curvature, separated by zones of high

11    curvature called *corners*. Corner dipoles are marked in the field array for use as described in

12    subsequent figures. Curvature can be determined by a variety of methods; in a preferred

13    embodiment, a dipole is considered a corner if its direction differs from that of either neighbor by

14    more than some specified parameter, e.g., as further described in conjunction with Fig. 8.

15    A sequence of zero or more propagate steps 262 extend the field out from the seeded positions.

16    The result is that force vectors pointing to pattern boundaries, as well as other information

17    needed by the run-time steps, can be obtained at some distance from the boundaries. The number

18    of propagate steps 262 is controlled by a parameter and determines the distance from pattern

19    boundaries that the field will contain valid force vectors, as well as the computation time needed

20    for pattern training. In a preferred embodiment, four propagation steps are used. Field elements

21    beyond the range of propagation will contain the code set during the initialization step 250.

1    Figure 3 shows a preferred embodiment of a feature detector to be used for practice of the

2    invention. The feature detector processes a source image 300. which can be either a training

3    image or a run-time image. A low-pass filter 310 and image sub-sampler 320 are used to

4    attenuate fine detail in the source image that for a variety of reasons we wish to ignore. For

5    example we may wish to attenuate noise or fine texture, or we may wish to expand the capture

6    range of the pattern by focusing on coarse pattern features.  Also, we may wish to decrease the

7    number of image dipoles, thereby reducing processing time.

8    The response of the filter 310 and sub-sampler 320 are controlled by parameters 220 stored in the

9    pattern (not shown in this figure). One setting of the parameters effectively disables the filter and

10   sub-sampler, allowing the source image 300 to pass without modification for maximum

11   resolution.

12   Methods for low-pass filtering and sub-sampling of digital images are well known in the art. In a

13   preferred embodiment, a constant-time second-order approximately Gaussian filter is used, as

14   described in [U.S. patent pending "Efficient, Flexible Digital Filtering"].

15   The filtered, sub-sampled image is processed by a gradient estimation module 330 to produce an

16   estimate of the x (horizontal) and y (vertical) components of image gradient at each pixel. A

17   Cartesian-to-polar conversion module 340 converts the x and y components of gradient to

18   magnitude and direction. A peak detection module 350 identifies points where the gradient

19   magnitude exceeds a noise threshold and is a local maximum along a 1-dimensional profile that

20   lies in approximately the gradient direction, and produces a list of the grid coordinates (row and

21   column number), gradient magnitude, and gradient direction for each such point.

1    A sub-pixel interpolation module 360 interpolates the position of maximum gradient magnitude

2    along said 1-dimensional profile to determine real-valued (to some precision) coordinates $(x_i, y_i)$

3    of the point. The result is a list of points that lie along boundaries in the image, which includes

4    the coordinates, direction, and magnitude of each point. This list can be used as the basis for

5    either a field or image dipole list, to which additional information may be added as appropriate.

6    Methods for identifying points along image boundaries are well-known in the art. Any such

7    method can be used for practice of the invention, whether based on gradient estimation or other

8    techniques. Methods for gradient estimation, Cartesian-to-polar conversion, peak detection, and

9    interpolation are also well-known. In a preferred embodiment, the methods described in [U.S.

10   patent pending "Method and Apparatus for Fast, Inexpensive, Subpixel Edge Detection"] are

11   used.

12   In a preferred embodiment, the source image has eight bits of gray-scale per pixel. The low-pass

13   filter produces a 16-bit image, taking advantage of the inherent noise-reduction properties of a

14   low-pass filter. The gradient estimation module uses the well-known Sobel kernels and operates

15   on either a 16-bit filtered image, if the parameters are set so as to enable the filter 310, or an 8-bit

16   unfiltered image if the parameters are set so as to disable the filter 310. The x and y components

17   of gradient are always calculated to 16 bits to avoid loss of precision, and the gradient magnitude

18   and direction are calculated to at least six bits preferably using the well-known CORDIC

19   algorithm.

20   Several parameter values are needed for feature extraction, both in the training module 110 and

21   in the run-time module 140. Generally these parameters include those controlling the response of

22   the low-pass filter 310, the sub-sampling amount used by sub-sampler 320, and the noise

1    threshold used by peak detector 350. Other values may be needed depending on the exact details

2    of the feature extractor used to practice the invention.

3    Appropriate settings for the parameter values depend on the nature of the patterns and images to

4    be analyzed. In a preferred embodiment certain defaults are used that work well in many cases,

5    but in general no rules can be given that work well in all cases. Said preferred embodiment is

6    - further described below in conjunction with figure 26.

7    Figure 4 shows an element of the field array as used in a preferred embodiment of the invention.

8    Information is packed into a 32-bit word 400, both to conserve memory and to speed up access

9    on conventional computers by maximizing the number of elements that will fit in data cache and

10   using a word size that keeps all elements properly aligned on appropriate address boundaries.

11   Fixed point representations are used for the force vector, both because they are more compact

12   than floating point representations and to allow best use to be made of the signal processing

13   capabilities of modern processors such as the Texas Instruments TMS320C80 and the Intel

14   Pentium-MMX.

15   In a preferred embodiment, a field element stores a force vector that gives the distance and

16   direction to the nearest point along a pattern boundary, and one bit that specifies whether the

17   gradient direction at that boundary point is in the same, or 180° opposite, direction as the force

18   vector. This is accomplished as shown in figure 4 by storing a signed force magnitude 410 and a

19   gradient direction 420. If the force direction is the same as the gradient direction, the force

20   magnitude 410 will be positive; if the force direction is opposite from the gradient direction, the

21   force magnitude 410 will be negative.

1 The magnitude/direction representation for the force vector is preferred over an x-y component

2 representation because it is necessary to be able to represent vectors that have zero length but a

3 well-defined direction. Such vectors are called *pseudo-null* vectors. The equivalent x-y

4 components can be calculated by the well-known formula

5
$$\begin{pmatrix} force_x \\ force_y \end{pmatrix} = magnitude \begin{pmatrix} \cos(direction) \\ \sin(direction) \end{pmatrix}$$

6 Note that gradient direction can be used in the above formula, since the stored magnitude is

7 negative if the force direction is opposite the gradient direction.

8 In the preferred embodiment shown, the force magnitude **410** is in units of field grid increments,

9 using a two's complement representation of 16 total bits, of which the least significant 11 are to

10 the right of the binary point and the most significant is the sign bit. Thus the maximum force

11 vector length is just under 16 field grid units, and the resolution is 1/2048$^{th}$ of a grid unit.

12 The gradient direction **420** is preferably represented as a 12-bit binary angle in the range 0° to

13 360°, with a resolution of 360°/4096 = 0.088°. In other embodiments, the bits of field element

14 400 are divided between force 410 and direction 420 to provide greater or lesser precision and

15 range, as needed for each particular application.

16 A 4-bit flags element **430** is also stored in the field element **400**. An 2-bit eval code **440**

17 determines how an image dipole is to be evaluated if the current pose maps it to a field position

18 within the region covered by this element 400. The *don't care* code specifies that the image

19 dipole should be ignored. The *expect blank* code specifies that no features are expected in this

20 region of the pattern, and so if any image dipoles map here they should be given a low evaluation

1     for inspection purposes, should be given a high clutter rating, and should not be used as evidence

2     for localization. The *evaluate only* code specifies that the image dipole should be evaluated by

3     the usual criteria for inspection purposes, but should not contribute evidence for localization

4     purposes. The *attract* code specifies that the image dipole should be evaluated and used both for

5     localization and inspection.

6     If the eval code 440 is either "don't care" or "expect blank", the force vector is undefined and is

7     said to be invalid. If the eval code is "evaluate only" or "attract", the force vector is said to be

8     valid.

9     A 1-bit corner code 450 specifies whether or not the pattern boundary point pointed to by the

10     force vector is in a high-curvature zone ("is corner") or a low-curvature segment ("not corner").

11     If the force vector is invalid, the corner code is set to "not corner".

12     A 1-bit polarity code 460 specifies whether the image dipole evaluation should consider or ignore

13     gradient direction, as described above in the summary section and further described below. A

14     parameter is used to specify whether or not to override the polarity flags stored in the field

15     element 400, and if so, whether to force polarity to be considered or ignored.

16     In a preferred embodiment, the field element 400 also specifies the identity of the nearest field

17     dipole 401 in addition to the force vector 400. The identity 401 can be represented as a index into

18     the field dipole list. In a preferred embodiment, a 16-bit index is used, which is stored in a

19     separate array so as to satisfy data alignment guidelines of conventional computers.

20     Figure 5 shows details of the initialization step 250 of the field generation step 210. A 2-

21     dimensional array 500 of field elements 400 is used. Any reasonable grid spacing can be used; in

1    a preferred embodiment, the grid spacing is the same as that of the image that is input to the

2    gradient estimation module 330 of the feature detector 200.

3    Field elements 520, (indicated as white in figure 5 and having the same structure as field element

4    400) cover the region of the training pattern 105, together forming a "training region". The field

5    elements 520 are initialized so that the eval code 440 is set to "expect blank". As described

6    above, in this state the force vector is considered invalid and need not be initialized. In one

7    embodiment, however, further described below in conjunction with figure 20b, the gradient

8    direction field 420 of these field elements 520 are set equal to the corresponding gradient

9    directions of the training image. A border of additional field elements 540, (indicated as gray in

10   figure 5 and having the same structure as field element 400) are initialized so that the eval code

11   440 is set to "don't care". This reflects the fact that in general we don't know what features might

12   lie beyond the bounds of the training region. These "don't care" values will be replicated inwards

13   during each propagation step 262, so that image features lying just outside the training pattern

14   105 don't attract to pattern features just inside.

15   A separate corresponding array of field dipole indices 401, identical in size to the white-shaded

16   field elements 520, is also used, but need not be initialized. The values in this array are

17   considered valid only if the force vector of the corresponding field element of array 500 is valid.

18   Figure 6 shows details of the seed step 252 of the field generation module 210. Shown is a subset

19   of the field elements 520 of the field array 500. Each field dipole is located within some field

20   element. For example, the field dipole at point 600 falls within field element 620, indicated as

21   gray in figure 6. Also shown is a small straight-line section of pattern boundary 660

1    corresponding to the example field dipole at point **600**. This section of boundary is shown

2    primarily to aid in understanding the figure. Its orientation, and position along a line normal to its

3    orientation, are significant, but its length is essentially arbitrary.

4    The field element **620** is set to have force vector **640**. The force vector points from the center of

5    element **620** to a point on boundary section **660** and either in the direction, or opposite to the

6  ·  direction of the dipole (i.e. normal to the boundary), whichever is required to bring the head of

7    the vector to the boundary **660**. In the example shown, the point on the boundary to which the

8    vector points is coincident with the dipole position **600**, but in general it need not be. Figure 6

9    shows several other examples of seeded force vectors.

10   It also may happen that a field dipole's position falls exactly at the center of a field element, so

11   that the length of the force vector is zero. In this case the force vector is pseudo-null — its

12   direction is well-defined and must be set properly.

13   In a preferred embodiment, for each field element that receives a seed force vector, the eval code

14   is set to "attract", the corner code is set to "no corner", and the polarity code is set to "consider

15   polarity". Other schemes may be devised within the spirit of the invention to suit specific

16   applications.

17   For each field element that receives a seed force vector, the corresponding element **401** of the

18   array of field dipole indices is set to identify the field dipole used to seed the field element.

19   In a preferred embodiment using the feature detector of figure 3, as further described in [U.S.

20   patent pending "Method and Apparatus for Fast, Inexpensive, Subpixel Edge Detection"], and

21   where the field grid has the same geometry as the image that is input to the gradient estimation

1    module 330, no more than one field dipole will fall within any given field element, and there will

2    be no gaps in the boundary due to grid quantization effects. In a less preferred embodiment using

3    different methods for feature detection, various schemes can be used to handle multiple dipoles

4    that fall within a given field element, or gaps in the boundary due to quantization effects. The

5    preferred method for multiple dipoles within one field element is to choose the one whose force

6    vector is shortest, and discard the others. The preferred method for gaps in the boundary is to do

7    nothing and let the propagation steps fill in the gaps.

8    Figure 7 shows details of the connect step 254 of the field generation module 210. Figure 7a

9    shows the same subset of field elements 520 of the field array 500 as was shown in figure 6. Also

10    shown is the example field element 620, indicated as light gray.

11    For every field dipole, the seeded field is examined to identify neighboring positions that contain

12    dipoles to which the dipole should be connected. For the example field element 620, the

13    neighboring positions 700 are shown, shaded medium gray. The neighboring positions 700 are

14    examined in two steps of four neighboring positions each, each step in a particular order,

15    determined by the direction of the field dipole corresponding to field element 620.

16    In one step, a left neighbor field element 710 is identified, and a left link 715 is stored in the field

17    dipole corresponding to field element 620 identifying the field dipole corresponding to field

18    element 710 as its left neighbor. In the other step, a right neighbor field element 720 is identified,

19    and a right link 725 is stored to identify the field dipole's right neighbor. If a given neighbor

20    cannot be found, a null link is stored. Note that "left" and "right" are defined arbitrarily but

21    consistently by an imaginary observer looking along the dipole gradient direction.

1    Figure 7b shows the order in which neighboring field elements are examined for a dipole whose

2    direction falls between arrows 740 and 742, corresponding to a pattern boundary that falls

3    between dotted lines 744 and 746. The sequence for identifying the left neighbor is +1, +2, +3,

4    and +4. The first neighbor in said sequence that contains a dipole (seeded field element), if any,

5    is the left neighbor. Similarly, the sequence for identifying the right neighbor is -1, -2, -3, and -4.

6    Figure 7c shows another example, where the dipole direction falls between arrows 760 and 762,

7    corresponding to a pattern boundary that falls between dotted lines 764 and 766. The sequences

8    of neighbors are as shown. The sequences for all other dipole directions are simply rotations of

9    the two cases of figures 7b and 7c.

10    Note that the sequences given in figures 7b and 7c show a preference for orthogonal neighbors

11    over diagonal neighbors, even when diagonal neighbors are "closer" to the direction of the

12    pattern boundary. This preference insures that the chains will properly follow a stair-step pattern

13    for boundaries not aligned with the grid axes. Clearly this preference is somewhat dependent on

14    the specific details of how the feature detector chooses points along the boundary.

15    Once connections have been established for all field dipoles, a consistency check is performed.

16    Specifically, the right neighbor of a dipole's left neighbor should be the dipole itself, and the left

17    neighbor of a dipole's right neighbor should also be the dipole itself. If any links are found for

18    which these conditions do not hold, the links are broken by replacing them with a null link. At

19    the end of the connect step, only consistent chains remain.

1     Many alternate methods can be used to connect dipoles within the spirit and scope of the

2     invention. In some embodiments, particularly where no inspection is to be performed, the

3     connect 254 is omitted entirely.

4     Figure 8 shows details of the segment step 260 of the field generation step 210. Field elements

5     800 shaded medium gray are identified as "corners" because the dipole directions differs from

6     that of their left and/or right neighbors by more than some specified parameter. In a preferred

7     embodiment, the parameter is 16.875 degrees. For these elements the corner code 450 is set to

8     "is corner". Other field elements 820, shaded light gray, lie along one chain segment, while field

9     elements 840, also shaded light gray, lie along another chain segment. For these elements it is not

10    necessary to set the corner code, because it was set to "no corner" when the field was seeded.

11    Figure 9 shows examples of part of the analysis that is performed by the propagate step 262 of

12    the field generation step 210. In the example of figure 9a, field element 900 initially does not

13    have a valid force vector; its eval code is "expect blank", as set by the initialization step 250.

14    Neighboring element 902 has a valid force vector 904, which points to a segment of pattern

15    boundary 906 that is assumed to be an approximately straight line.

16    A vector 908 is constructed from the center 916 of element 900 to the center of the neighbor 902.

17    The projection 910 of vector 908 onto force vector 904 is constructed. A new force vector 912 is

18    constructed from the center 916 of field element 900 to the boundary 906 by adding the

19    neighbor's force vector 904 to the projection 910. An offset value is computed whose magnitude

20    is equal to the length 914 of the difference between vector 908 and projection 910, and whose

21    sign is determined by the direction 918 by which vector 908 must be rotated to coincide with

1    projection 910, where anti-clockwise is positive as shown and clockwise is negative. The result

2    of this analysis is the new force vector 912 and offset value of magnitude 914 and sign 918.

3    A similar example but for a diagonal neighbor 932 of element 930 is shown in figure 9b. The

4    projection 940 of vector 938 onto force vector 934 is constructed. A new force vector 942 is

5    constructed from the center 946 of field element 930 to the boundary 936 by adding the

6    `neighbor's force vector 934 to the projection 940. An offset value is computed whose magnitude

7    is equal to the length 944 of the difference between vector 938 and projection 940, and whose

8    sign is negative since vector 938 must be rotated clockwise 948 to coincide with projection 940.

9    Another example is shown in figure 9c, where in this case the boundary 966 passes between field

10   element 960 and its neighbor 962. The projection 970 of vector 968 onto force vector 964 is

11   constructed. A new force vector 972 is constructed from the center 976 of field element 960 to

12   the boundary 966 by adding the neighbor's force vector 964 to the projection 970. An offset

13   value is computed whose magnitude is equal to the length 974 of the difference between vector

14   968 and projection 970, and whose sign is negative since vector 968 must be rotated clockwise

15   978 to coincide with projection 970.

16   Further details of propagate step 262 of the field generation step 210 are shown in figure 10.

17   Each element of the field array is examined. Any element whose eval code 440 is "expect blank"

18   is considered for possible propagation of the field to that element. All other field elements are

19   already in a final state and are skipped. For each field element so considered, the eight neighbors

20   are examined. If two or more adjacent neighbors have valid force vectors or have eval codes

1     equal to "don't care", the field will be propagated to the said field element; otherwise, the field

2     element will be skipped and possibly considered again on a subsequent propagate step.

3     The rule specifying two or more adjacent neighbors is used to insure that there is sufficient

4     information to be able to interpolate the field between neighbors. "Adjacent" means either

5     sharing an edge, such as elements **1010** and **1012** of figure 10, or sharing a corner, such as

6     - elements **1010** and **1014**.

7     In figure 10 element **1000** shaded light gray has eval code "expect blank", and neighbors **1010**,

8     **1012**, and **1014**, shaded medium gray, have valid force vectors (the field has been seeded at or

9     already propagated to the neighbors). Neighboring element **1010** has force vector **1030**, and

10     following the method of figure 9 new force vector **1032** and positive offset **1034** are computed.

11     Neighboring element **1012** has force vector **1040**, and following the method of figure 9 new force

12     vector **1042** and negative offset **1044** are computed. Neighboring element **1014** has force vector

13     **1050**, and following the method of figure 9 new force vector **1052** and negative offset **1054** are

14     computed.

15     The neighbors of field element **1000** are scanned anti-clockwise in sequence **1070**. The starting

16     and ending points of sequence **1070** are arbitrary. If exactly one positive to negative offset

17     transition between adjacent neighbors is found, the field is propagated to element **1000** by

18     constructing a force vector **1080** by interpolating between new force vectors **1032** and **1042**.

19     In a preferred embodiment, the interpolation is a weighted average of vectors **1032** and **1042**.

20     The vector **1032** is weighted by the magnitude of offset **1044**, and the vector **1042** is weighted by

21     the magnitude of offset **1034**. The effect is that the weight of a vector is proportional to the other

1    vector's offset and inversely proportional to its own offset, so that vectors are more heavily

2    weighted if they pass closer to their corresponding neighbor's force vector. As shown in figure

3    10, the offset corresponding to vector **1032** has the smaller magnitude, so it is more heavily

4    weighted and therefore force vector **1080** passes closer to vector **1032** than to vector **1042**.

5    One method for constructing a weighted average of vectors is to scale each vector by its

6    · corresponding weight, add the results, and then scale by the inverse of the sum of the weights.

7    This is equivalent to an independent weighted average of the x and y components. In a preferred

8    embodiment, an independent weighted average of the magnitude and direction is used.

9    If the vectors **1032** and **1042** participating in the interpolation are pointing to distant points along

10    a pattern boundary, or to different boundaries, the field is considered indeterminate at element

11    **1000** and the eval code is set to "don't care". In a preferred embodiment, the vectors are

12    considered to be pointing to distant points or different boundaries if either their magnitudes differ

13    by more than 3 grid units or their directions differ by more than 135°.

14    In a preferred embodiment, a special case rule is used to propagate the field at the ends of an

15    open chain. If a neighboring element with a valid force produces a small positive offset, and no

16    anti-clockwise adjacent neighbor has a valid force, the field will propagate without interpolation

17    by using the new force vector as constructed by the method of figure 9. Similarly, if a

18    neighboring element with a valid force produces a small negative offset, and no clockwise

19    adjacent neighbor has a valid force, the field will propagate without interpolation by using the

20    new force vector as constructed by the method of figure 9. In a preferred embodiment, a small

21    offset is one whose magnitude is less than $1/10^{th}$ of a grid unit.

1  If more than one positive to negative offset transition between adjacent neighbors, or application

2  of the special case rule, is found, or if none are found, the field is considered indeterminate at

3  element **1000** and the eval code is set to "don't care". One reason that no such transitions might

4  be found is that neighboring field elements are themselves set to "don't care", for example the

5  border elements **540** set by the initialization step **250**.

6  - If a valid force is propagated to element **1000**, then corner code **450**, polarity code **460**, and the

7  index of the nearest field dipole are also propagated by copying from whichever of the

8  neighboring elements participating in the interpolation has the smallest offset magnitude (greatest

9  weight). In the example of figure 10, the values would be copied from element **1010**. If the

10  special case rule was applied, the values are copied from the neighbor with small offset used to

11  construct the new force vector.

12  Many variations on the above rules can be used within the spirit of the invention to achieve

13  similar results. Indeed any method that produces force vectors that point to the nearest point

14  along a pattern boundary can be used to practice this invention.

15  Figure 11 shows the same portion of the field array that was shown after seeding in figure 6, but

16  with new force vectors resulting from one propagation step. Figure 12 shows the same portion

17  after two propagation steps. Note in figure 12 field element 1200 whose eval code is set to "don't

18  care" because more than one positive to negative offset transition between adjacent neighbors

19  was found.

20  Figure 13 shows a block diagram of the run-time module **140** of a preferred embodiment. Run-

21  time module **140** analyzes the image **130**, using the stored pattern **120**, the starting pose 132, and

1    the client map 131. As a result of the analysis, the run-time module produces a pose 134 that

2    maps pattern points to accurately corresponding image points.

3    The run-time module 140 produces an *rms error* value 136 that is a measure of the degree of

4    match between the pattern and the image, a *coverage* value 138 that is a measure of the fraction

5    of the pattern to which corresponding image features have been found, and a clutter value 137

6  . that is a measure of extra features found in the image that do not correspond to pattern features.

7    The run-time module 140 produces an evaluated image dipole list 150 and an evaluated field

8    dipole list 160, e.g., as shown in Fig. 25. The clutter values of the evaluated image dipole list 150

9    can be used to identify features in the image 130 not present in the pattern 105 (shown in figure

10   1). These probability values range from 0 to 1 and indicate the likelihood that the image feature

11   is not present in the pattern. The "eval2" values (Fig. 25) of the evaluated field dipole list 160

12   can be used to identify features in the pattern 105 not present in the image 130. These

13   probability values range from 0 to 1, and indicate the likelihood that the pattern feature was

14   found in the image.

15   The run-time module 140 uses a feature detection module 200 to process the image 130 to

16   produce an image dipole list 1300. In a preferred embodiment, the feature detection module 200

17   is identical to that used by training module 110, and is controlled by the same parameter settings

18   stored in pattern parameters 220, and is further described in conjunction with Fig. 26. In other

19   embodiments, different methods or different parameters setting are used as appropriate for a

20   specific application.

1    At least one attraction module 1350 uses pattern parameters 220, field dipole set 230, field 240,

2    image dipole list 1300, and the starting pose 132 and client map 131, to refine the starting pose

3    132 and produce the other outputs 136, 138, 160, and 150.

4    Figure 14 is a diagram that is used to derive the mathematical basis for a preferred embodiment

5    that uses a least-squares method to determine a pose that best accounts for the evidence of the

6  ·  image dipoles at each attraction step. An image dipole, mapped by the current pose to field point

7    1400 and with direction 1402, is considered. A small section of image boundary 1404 is also

8    shown as an aid in understanding the diagram.

9    The field has force f 1430 at mapped image dipole point 1400, pointing to the nearest point 1435

10   along pattern boundary 1410. Note that the force 1430 is normal to the pattern boundary 1410 at

11   point 1435, and the mapped image dipole direction 1402 is similar but not equal, modulo 180°,

12   to that of the force.

13   The mapped image dipole point 1400 has position vector p 1450 relative to the field origin 1420.

14   The existence of an image dipole at field point 1400, with force 1430 and mapped dipole

15   direction similar to the force direction (in a preferred embodiment, similar modulo 180°), is

16   taken as evidence that the current pose should be modified so that the image dipole is mapped so

17   as to lie somewhere along line 1440 tangent to pattern boundary 1410 at point 1435. The dipole

18   provides no evidence as to position normal to the force, that is along tangent 1440. The position

19   vector p' 1470 defines a point on tangent 1440, and the difference vector p - p' 1460 indicates

20   how the mapped dipole position might move as a result of the force.

1     Suppose that [C, t] is a six-degree-of-freedom coordinate transform that maps the current pose

2     into a new, hopefully more accurate, pose. This transform is called the *motion* transform, because

3     it tells how the image dipoles will move with respect to the field under the influence of the forces

4     of the field. Here C is a 2x2 matrix and t is a translation vector. The evidence under

5     consideration suggests that this transform should map **p** to **p'**:

6                                 $$\mathbf{p'} = C\mathbf{p} + \mathbf{t}. \tag{1}$$

7     Let **I** be the identity matrix and define

8                                   $$f = |\mathbf{f}| \tag{2}$$

9                                   $$\hat{\mathbf{f}} = \frac{\mathbf{f}}{|\mathbf{f}|} \tag{3}$$

10                                 $$\dot{C} = C - I \tag{4}$$

11     From the diagram of figure 14 it can be seen that

12                                   $$f = (\mathbf{p'} - \mathbf{p}) \cdot \hat{\mathbf{f}} \tag{5a}$$

13                                   $$= (C\mathbf{p} + \mathbf{t} - \mathbf{p}) \cdot \hat{\mathbf{f}} \tag{5b}$$

14                                   $$= [(C - I)\mathbf{p} + \mathbf{t}] \cdot \hat{\mathbf{f}} \tag{5c}$$

15                                   $$= (\dot{C}\mathbf{p} + \mathbf{t}) \cdot \hat{\mathbf{f}} \tag{5d}$$

1    Thus, given an image dipole that maps to field point $\mathbf{p}$ with force $\mathbf{f} = f\hat{\mathbf{f}}$, we have one equation

2    in the six unknowns $[\mathbf{C}, \mathbf{t}]$ that tells us how to map the current pose to get a new pose. With six

3    dipoles we can solve for the six-degrees-of-freedom, but in practice the evidence obtained from

4    only six dipoles is generally not sufficient to get an accurate or even meaningful solution. In

5    practice we use many dipoles, typically anywhere from a few dozen to a few thousand, and some

6    method for solving an over-determined set of equations.

7    In a preferred embodiment, a least-squares method is used. An error term for the $i^{th}$ dipole can be

8    defined as

9
$$e_i = (\dot{\mathbf{C}}\mathbf{p}_i + \mathbf{t}) \bullet \hat{\mathbf{f}}_i - f_i \tag{6}$$

10    With this definition a least-squares problem can be set up and solved by methods well-known in

11    the art. If weights $w_i$ are determined for each dipole, we can write the sum squared error as

12
$$E = \sum_i w_i \left[ (\dot{\mathbf{C}}\mathbf{p}_i + \mathbf{t}) \bullet \hat{\mathbf{f}}_i - f_i \right]^2 \tag{7}$$

13    In practice it is usually desirable to solve for fewer than six-degrees-of-freedom. Some patterns

14    would result in a singular or unstable solutions if certain degrees of freedom are included. For

15    example, circles cannot be solved for orientation and corners cannot be solved for size. In other

16    cases, a solution would be possible, but some degrees of freedom, particularly aspect ratio and

17    skew, are known not to vary and might cause problems if included. Perhaps the most serious

18    such problem is that unreliable evidence, present to some degree in all images, will have a more

19    serious effect when more degrees of freedom are allowed to vary. Another problem is that

20    somewhat more computation is needed to solve for the additional degrees of freedom.

1   In a preferred embodiment the least-squares problem is set up in 4 degrees of freedom

2   corresponding to x translation, y translation, orientation, and size. Sums needed for a least-

3   squares solution in the 4 degrees of freedom are computed, and pattern parameters 220 specify

4   which of the degrees of freedom will be solved for.

5   In an orthonormal coordinate system we can constrain the matrix $\dot{C}$ to orientation and size

6   . variation by writing it as

$$\dot{C} = \begin{pmatrix} p & q \\ -q & p \end{pmatrix} \tag{8a}$$

$$= pI + q\begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \tag{8b}$$

9   In practical applications, however, the images themselves are almost never orthonormal. CCD

10   cameras, for example, typically have pixels that are non-square by a percent or so. For line scan

11   cameras, the angle between the coordinate axes depends on mechanical alignment and so the

12   coordinate axes may not be perfectly orthogonal. The variations from square are small, but easily

13   detectable given the accuracy that can be achieved with the invention. Furthermore, it is

14   sometimes useful to have a significantly non-orthonormal field. For example, a field generated

15   from a square pattern can be used to localize and inspect a rectangular or even parallelogram-

16   shaped instance of the pattern by using an appropriate starting pose.

17   In these cases we generally want the orientation degree of freedom defined by an orthonormal,

18   real-world coordinate system rather than image or field coordinates. We re-write $\dot{C}$ as

1 $$\dot{\mathbf{C}} = p\mathbf{I} + q\mathbf{N} \qquad (9)$$

2 where the elements of matrix $\mathbf{N}$ are the components of the *normal tensor* in the field coordinate

3 system. The normal tensor is a mixed $2^{nd}$-rank tensor, a vector-valued function of vectors that,

4 informally, tells how to rotate a vector 90°. In an orthonormal coordinate system, of course, the

5 components of the normal tensor are

6 $$\begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}.$$

7 The components of the normal tensor are computed from the current pose and from a coordinate

8 transform called the *client map* that transforms points in an orthonormal but otherwise arbitrary

9 reference coordinate system to points in the run-time image.

10 We can now re-write equation 7, the sum squared error, as

11 $$E = \sum_i w_i \left[ \left[ (p\mathbf{I} + q\mathbf{N})\mathbf{p}_i + \mathbf{t} \right] \bullet \hat{\mathbf{f}}_i - f_i \right]^2 \qquad (10a)$$

12 $$= \sum_i w_i \left[ p\mathbf{p}_i \bullet \hat{\mathbf{f}}_i + q\mathbf{N}\mathbf{p}_i \bullet \hat{\mathbf{f}}_i + \mathbf{t} \bullet \hat{\mathbf{f}}_i - f_i \right]^2 \qquad (10b)$$

13 Now we can substitute

14 $$r_i = \mathbf{p}_i \bullet \hat{\mathbf{f}}_i \qquad (11)$$

15 $$s_i = \mathbf{N}\mathbf{p}_i \bullet \hat{\mathbf{f}}_i \qquad (12)$$

$$\mathbf{t} = \begin{pmatrix} x \\ y \end{pmatrix} \qquad (13)$$

$$\hat{\mathbf{f}}_i = \begin{pmatrix} u_i \\ v_i \end{pmatrix} \qquad (14)$$

into equation 10b and finally we have

$$E = \sum_i w_i \left[ (xu_i + yv_i + pr_i + qs_i - f_i) \right]^2 \qquad (15)$$

A least-squares problem based on equation 15 is easy to set up and solve for $x$, $y$, $p$, and $q$ by

well-known methods. The desired motion transform [C, t] is obtained from said solution using

equations 4, 9, and 13. The current pose is composed with the motion transform to obtain the

new pose.

Figure 15 is a block diagram of the attraction module 1350 for a preferred embodiment based on

a least-squares method of best accounting for the evidence of the image dipoles. In addition, to

further clarify a preferred sequence of operation of the modules of Fig. 15, a flow chart is

provided in Figs. 27A and 27B. Steps of the flow chart include reference numbers from Fig. 15

in parentheses to help cross-correlate the figures. A current pose 1500 is initially set to the start

pose 2702 and updated at the end of each attraction step 2724. After the sum module 1535 is

initialized to zero 2704, the Normal tensor computation module 1510 uses the current pose and

client map to compute the normal tensor N 2706 for the current attraction step.

Each image dipole 1515 of the image dipole list 1300 (figure 13) is processed 2708. The position

and direction of dipole 1515 are mapped 2710 from image coordinates to field coordinates by

1  map module 1520, using the current pose 1500. The mapped position is used by field module

2  1525 to determine the force, flags 430, and index of nearest field dipole 2712. The force, flags,

3  and index are stored in the image dipole 1515 for later use. The normal tensor, force, and image

4  dipole position in field coordinates are used by a rotate module 1530 to obtain the dipole's

5  position in force coordinates $(r, s)$ 2714 as specified by equations 11 and 12.

6  ·An evaluate module 1545 examines the force, flags, image dipole direction in field coordinates,

7  and dipole gradient magnitude and computes a weight for attraction (localization) purposes and

8  evaluation and clutter values for inspection purposes 2714. In some embodiments, the evaluate

9  module 1545 also considers the rms error from the previous attraction step in determining the

10  weight and evaluation. The evaluation and clutter values are stored in the image dipole 1515 for

11  later use.

12  A sum module 1535 uses the force, dipole position in force coordinates, and weight to compute

13  sums needed for the least-squares solution 2716. If there are no more dipoles 2718, a solve

14  module 1540 uses the sums and the normal tensor to solve for the motion transform and compute

15  the rms error 2720. A compose module 1505 composes the current pose with the motion

16  transform to produce a new pose 2722, which will be the current pose for the next attraction step,

17  or the final pose if this is the last attraction step 2726.

18  In a preferred embodiment where inspection is being performed, at the end of the last attraction

19  step, the field dipole evaluation module 1550 evaluates the image dipole list 2728, which has

20  now been evaluated by evaluate module 1545, and the field dipole set 230 stored in the pattern

21  120, and produces an evaluated field dipole list, coverage rating, and clutter rating 2730.

1  Figure 16 gives details for the map module 1520 of figure 15. Inputs are image dipole position in

2  image coordinates 1600, image dipole direction with respect to image coordinates 1610, and the

3  current pose 1620. One output is the dipole position in field coordinates 1630, computed as

4  shown.

5  The other output is the dipole direction with respect to field coordinates 1640, computed as

6  shown. The formula for output dir(field) 1640 effectively does the following, reading the vector

7  and matrix operations right to left:

8       • Construct a unit vector in the dipole direction, with respect to image coordinates,

9         by computing the cosine and sine of the angle $\theta_d$.

10      • Rotate the unit vector 90° to get a direction along the boundary that contains the

11        dipole.

12      • Map the rotated unit vector to field coordinates using $C_p$ to get a boundary

13        direction in field coordinates.

14      • Rotate the mapped rotated unit vector -90° to get a direction normal to the

15        boundary in field coordinates.

16      • If the determinant of the pose matrix is negative, the transform changes the left- or

17        right-handedness of the coordinate system, so rotate the vector 180° because the -90°

18        of the previous step should have been +90°.

1       • Compute the angle of the resulting vector using the well-known version of the

2       arctangent function of two arguments whose result is in the range 0° to 360°.

3   Note as shown in output dir(field) **1640** that these calculations can be simplified considerably. In

4   a preferred embodiment, the simplified formula is used at the beginning of each attraction step to

5   compute a 256-element lookup table, indexed by an 8-bit binary angle, for use by the map block

6   `1520`. This allows the direction mapping operation to be executed at high speed for each dipole.

7   When computing the lookup table, the symmetry of the formula requires us to compute only 128

8   elements of the table; the other elements are the negative of the computed ones. As a further

9   improvement in computation time, 64 even-indexed elements are computed, and the odd-indexed

10  elements are determined by interpolation from the even-indexed elements. Thus the formula need

11  only be applied 64 times. The arctangent function is computed using the well-known CORDIC

12  method.

13  Note that in computing output dir(field) **1640** we map the boundary direction instead of the

14  dipole direction. This is because, in the embodiment described herein, directions are determined

15  by a gradient estimation method **330** and Cartesian to polar conversion method **340** that assumes

16  square pixels. This is not a problem except when mapping directions between non-orthonormal

17  coordinate systems. In that case, the boundary direction must be used.

18  Figure 17 shows a block diagram of the field module **1525** of figure 15, and a corresponding

19  geometric diagram that illustrates the computation being performed. Image dipole position in

20  field coordinates **1754** is input to the field block **1525**. The coordinates **1754** fall within field grid

21  cell **1750**.

1    The coordinates are rounded to integer field grid position **1758** by integer rounding module **1700**.

2    The integer field grid coordinates **1758** are used by address generation module **1704** to produce a

3    memory address used to look up field element **1708**, corresponding to grid cell **1750**. In the

4    embodiment shown, the index of the nearest field dipole is stored with the other field

5    information, but in some embodiments, as described herein, the index is kept in a separate array.

6    The force direction $\theta_f$, flags, and index obtained from field element **1708** are direct outputs of

7    field module **1525**, but the force magnitude **1774** is interpolated so that the force vector is a

8    reasonably smooth function of real-valued position within the field.

9    Force interpolation is based on the assumption that the force stored in field element **1708**,

10   corresponding to integer grid position **1758**, points to an approximately straight-line section of

11   pattern boundary **1782**. This is a fast and accurate interpolation we can use with the information

12   available. A more compute intensive interpolation could use neighboring field elements as well.

13   To interpolate force magnitude, the integer position **1758** is subtracted by module **1712** from the

14   real-valued position **1754** to produce sub-grid position vector **1762**. A unit vector **1766** in the

15   force direction is constructed by cosine/sine module **1716**, implemented as a lookup table in a

16   preferred embodiment. The dot product **1770** of sub-grid position vector **1762** and unit vector

17   **1766** is computed by dot product module **1720**. The dot product **1770** is subtracted from force

18   magnitude **1774** by the subtraction module **1724** to produce interpolated force magnitude **1778**.

19   The interpolated force magnitude **1778**, unit vector in the force direction **1766**, and force

20   direction angle stored in field element **1708**, are collected in output module **1728** and become

21   part of the force vector **1526** produced by field module **1525**.

1    In another embodiment, not shown, at least one force vector is stored in each field element,

2    pointing to the nearest points along at least one pattern boundary. The field module 1525

3    examines image dipole direction in addition to position, and uses the stored force vector that is

4    closest to the dipole direction for interpolation and output to subsequent steps.

5    Figure 18 gives details for the rotate module 1530 of figure 15. Inputs are the normal tensor

6    1800, force 1810, and image dipole position in field coordinates 1820. Output is image dipole

7    position in force coordinates pos(force) 1830, computed as shown, and as described above by

8    equations 11 and 12.

9    Figure 19 shows various preferred fuzzy logic processing modules that are used in evaluate

10   module 1545 of a preferred embodiment illustrated in Fig. 20.

11   Figure 19a shows a *fuzzy greater than* module, which takes a real-valued input 1900 and a fuzzy

12   threshold 1904, and produces a fuzzy logic value 1912. The fuzzy threshold 1904 is an ordered

13   pair that specifies points along the x axis of graph 1908. The graph 1908 shows the fuzzy logic

14   output 1912 as a function of input 1900. As can be seen, the fuzzy logic value falls within the

15   range 0.0 to 1.0, inclusive.

16   Figure 19b shows a *fuzzy less than* module, which takes a real-valued input 1930 and a fuzzy

17   threshold 1934, and produces a fuzzy logic value 1942. The fuzzy threshold 1934 is an ordered

18   pair that specifies points along the x axis of graph 1938. The graph 1938 shows the fuzzy logic

19   output 1942 as a function of input 1930. As can be seen, the fuzzy logic value falls within the

20   range 0.0 to 1.0, inclusive.

1 Figure 19c shows a *fuzzy not* module. Fuzzy logic value input 1960 is inverted by subtracting it

2 from 1 to produce fuzzy logic value output 1964.

3 Figure 20 is a block diagram of a preferred embodiment of the evaluate module 1545 of figure

4 15. Figure 20a shows the portion responsible for computing the *weight* and *eval* values, and

5 figure 20b shows the portion responsible for computing the *clutter* value.

6 Referring to figure 20a, the computation of *weight* and *eval* is based on the force magnitude 'f', a

7 comparison of the force direction $\theta_f$, and image dipole direction dir(field), and the image dipole's

8 gradient magnitude 'mag'. For each of these three factors in the evaluation, a fuzzy logic value is

9 produced by fuzzy logic modules 2004, 2040, 2064, respectively, that indicates confidence in the

10 reliability of the evidence provided by the image dipole being evaluated. The three fuzzy

11 confidence factors so-produced are combined into a single confidence score by a combination

12 module 2080 in the range 0.0 to 1.0. The *weight* and *eval* outputs are obtained by using the eval

13 code 440 (figure 4) to select either the confidence score or the value 0.0.

14 Absolute value module 2002 computes the length of the force vector from force magnitude *f*,

15 which in the preferred embodiment being described may be negative if the force and gradient

16 directions differ. Fuzzy less than module 2004 compares the force length to a field strength

17 threshold 2000, to produce a strength confidence factor that indicates high confidence for force

18 lengths "below" the field strength threshold.

19 The field strength threshold 2000 is set based on pattern parameters 220 for the first attraction

20 step. In a preferred embodiment, the first attraction step uses field strength threshold values $t_{zero} =$

21 2.0 field grid units, and $t_{one} = 3.0$ field grid units.

1    In the embodiment shown in figure 20, the field strength threshold 2000 is modified after each

2    attraction step based on the rms error from the previous step. The modification is accomplished

3    by addition module 2012, which adds the rms error to both the $t_{zero}$ and $t_{one}$ components of a field

4    strength margin parameter 2008 to produce the new field strength threshold 2000. As a result, the

5    field strength threshold 2000 is matched to how well the particular run-time image being

6    analyzed fits the stored pattern at each attraction step.

7    The method of adjusting the field strength threshold based on the rms error of the previous step is

8    effective in some applications, but in other cases it has been observed to result in some

9    oscillation of the attraction rather than convergence on one solution. In a preferred embodiment,

10   not shown, the field strength threshold is reduced in equal steps after each attraction step. Thus,

11   as the attraction converges to a solution, image dipoles must be closer to pattern boundaries to be

12   given high confidence.

13   The image dipole direction dir(field) is compared with the pattern boundary gradient direction by

14   subtract module 2024. Recall that in the embodiment being described, the "force" direction $\theta_f$

15   reported by the field is actually boundary gradient direction, which is the same as or opposite of

16   the true force direction. If pattern parameters 220 and polarity code 460 of flags 430 specify that

17   gradient polarity is to be ignored, the angle difference from subtract module 2024 is constrained

18   to the range −90° to +90° by mod 180° module 2028; otherwise, the angle $\theta_f$ is passed

19   unmodified. The magnitude of the resulting angle difference is determined by absolute value

20   module 2032.

1    The angle difference magnitude is compared to one of two fuzzy thresholds by fuzzy less than

2    module 2040 to produce a direction confidence factor. If corner code 450 of flags 430 indicates

3    "no corner", the field direction threshold 2044 is chosen by selection module 2036. If the corner

4    code indicates "is corner", the field corner threshold 2048 is chosen. In a preferred embodiment,

5    the field direction threshold 2044 has values $t_{zero} = 11.25°$ and $t_{one} = 22.5°$, and the field corner

6    threshold 2048 has values $t_{zero} = 39.375°$ and $t_{one} = 50.625°$, reflecting the fact that a wider range

7    of image dipole directions can reasonably correspond to a pattern boundary corner. In an

8    alternate embodiment, a real-valued measure of curvature can be used instead of the binary "is

9    corner" code, with multiple values of the field direction threshold possible.

10   The image dipole's gradient magnitude 'mag' is compared to a fuzzy magnitude threshold 2060

11   by fuzzy greater than module 2064 to produce a magnitude confidence factor. The magnitude

12   threshold is intended to throw out very weak dipoles that are likely due to image noise or other

13   artifacts, but the use of a fuzzy threshold gives more stable results than the more traditional hard

14   threshold. In a preferred embodiment, the magnitude threshold 2060 uses the same value for $t_{zero}$

15   as the noise threshold chosen for the peak detector 350, and uses a value of $t_{one}$ equal to twice the

16   value of $t_{zero}$.

17   The strength, direction, and magnitude confidence factors are combined by multiply module

18   2080 to produce an overall confidence score in the range 0 to 1. Based on eval code 440 of flags

19   430, the selection module 2084 chooses a value for *weight* and the selection module 2088

20   chooses a value for *eval*. If the eval code is "attract", the confidence score is chosen for the

21   weight; otherwise the constant 0 is chosen so that the dipole is ignored for localization purposes.

22   If the eval code is "attract" or "evaluate only", the confidence score is chosen for *eval*; otherwise

1     the constant 0 is chosen to indicate that the dipole does not correspond to any portion of the

2     pattern.

3     Figure 20b shows a preferred embodiment for the calculation of the *clutter* value. The direction

4     confidence factor produced by fuzzy less than module **2040** is inverted by fuzzy not element

5     **2042**. The image dipole's gradient magnitude is compared to a fuzzy clutter threshold **2070** by

6     . fuzzy greater than module **2074** to produce a clutter confidence factor 2075. The clutter

7     confidence factor 2075 is multiplied by the inverted direction confidence factor by multiplier

8     **2090** to produce a tentative clutter value 2091. If the eval code **440** is anything but "don't care",

9     selection module **2092** chooses this tentative clutter value 2091 as the dipole's *clutter* value;

10     otherwise the constant 0 is chosen.

11     If the eval code 440 is "expect blank", the force magnitude 'mag' is meaningless, but in a

12     preferred embodiment, the force direction $\theta_f$ encodes the gradient direction from the training

13     image as described above in conjunction with figure 5. In this case, the computation of clutter

14     uses this direction as it would the force direction $\theta_f$. This mode of operation is appropriate when

15     it is desirable to minimize false alarms. Alternatively, if it is appropriate to minimize the chances

16     of missing clutter, one can consider only the clutter confidence factor 275 when the eval code is

17     "expect blank".

18     In a preferred embodiment, the clutter threshold **2070** has values of $t_{zero}$ and $t_{one}$, each value being

19     equal to 1.5 times the $t_{zero}$ and $t_{one}$ values used for the magnitude threshold **2060**.

20     In a preferred embodiment, the magnitude confidence factors and clutter confidence factors for

21     all of the image dipoles are computed once and stored in the image dipole list **1300**, rather than

1    being recomputed for each attract step. This can be done because these confidence factors are

2    independent of the current pose **1500.**

3    Figure 21 is a block diagram of sums module **1535** of figure 15. This module accumulates the

4    weighted sums needed for the solution of the least-squares problem of equation 15. Five multiply

5    modules **2100** perform the weighting. Fifteen multiply-accumulate modules **2130** and one

6    accumulate module **2160** compute and store the sums needed. The sixteen accumulators are set

7    to zero at the beginning of each attraction step. This four degree-of-freedom case is an

8    exemplary embodiment, other numbers of degrees of freedom being possible.

9    Figures 22 a-d give details of the solve module **1540** of figure 15, which produces the motion

10   transform and the rms error value. The formulas shown are based on the solution of the least-

11   squares problem of equation 15. Pattern parameters **220** specify which degrees of freedom are to

12   be determined.

13   Figure 22a shows the solution for the 2 translation degrees of freedom only — size and

14   orientation are as specified in the start pose. Figure 22b shows the solution for translation and

15   orientation. This preferred solution is based on an approximation that assumes a small angle of

16   rotation. If the assumption is violated, some size variation will be introduced. Figure 22c shows

17   the solution for translation and size, holding orientation fixed, and figure 22d shows the solution

18   for all 4 degrees of freedom.

19   Figure 23 gives details of the compose module **1505** of figure 15, which composes the current

20   pose **2300** with the motion transform **2330** computed by the solve module **1540** to produce the

21   new pose **2360.**

1    Figure 24 gives details of the normal tensor module 1510 of figure 15, which computes the

2    normal tensor 2460 from the current pose 2400 and the client map 2430.

3    Figure 25 shows an example of field dipole evaluation performed as part of field dipole

4    evaluation module 1550 of figure 15. In the example, a first image dipole 2500, second image

5    dipole 2510, and third image dipole 2520 have received evaluations 0.85, 0.93, and 0.88

6    · respectively. Four field dipoles labeled 2540, 2550, 2560, and 2570 lie along a chain as

7    determined by connect step 254 during training module 110. The chain is defined by the left links

8    2580 and right links 2585.

9    For image dipole 2500, an index 2505 was determined by field module 1525 to identify the

10   nearest field dipole 2540. The evaluation 0.85 is transferred from image dipole 2500 to the

11   "eval1" slot of field dipole 2540.

12   No image dipole identified field dipole 2550 as nearest, so its "eval1" slot holds its initial value

13   0.

14   For image dipole 2510, an index 2515 was determined to identify the nearest field dipole 2560.

15   For image dipole 2520, the same index 2515 was determined to identify the nearest field dipole

16   2560. The larger of image dipole 2510 evaluation 0.93 and image dipole 2520 evaluation 0.88 is

17   transferred to the "eval1" slot of field dipole 2560.

18   Field dipole 2570 has evaluation 0.90 transferred from some image dipole not shown.

19   To fill in the gap at field dipole 2550, a dilation operation is performed, wherein all field dipoles

20   receive an evaluation equal to the maximum of their own evaluation and that of their left and

1  right neighbors. The dilated evaluations are shown in the "eval2" slot of each field dipole. Note

2  that it is not actually necessary to store both "eval1" and "eval2" values; figure 25 shows them

3  for clarity.

4  Once the field dipoles have been evaluated, the coverage value produced by field dipole

5  evaluation module 1550 is computed by averaging all of the field dipole evaluations.

6  In a preferred embodiment, the field dipoles are evaluated and coverage is computed only after

7  the last attraction step, and only if pattern inspection is desired.

8  Figure 26 shows how the invention can be operated in a multi-resolution mode designed to

9  increase the capture range without sacrificing accuracy. Two pattern training modules (not

10  shown) are run on a single training image, with different settings of low-pass filter module 310

11  and image sub-sample module 320. In a first setting designed to attenuate fine detail, a low

12  resolution pattern 2600 is generated. In a second setting designed to pass fine detail, a high

13  resolution pattern 2610 is generated.

14  A low resolution run-time module 2620 uses the low resolution pattern 2600, and a start pose and

15  client map, to analyze run-time image 130 to produce a low resolution pose that is much more

16  accurate than the start pose but not as accurate as can be achieved at higher resolution. A high

17  resolution run-time module 2630 uses the high resolution pattern 2610, the low resolution pose

18  as a start pose, and the same client map, to analyze run-time image 130 to produce the final pose,

19  rms error, coverage, and evaluated dipole lists.

20  The multi-resolution mode is supervised by overall control module 2640, as illustrated by the

21  flow chart in Fig. 28. As part of its operation 2800, the low resolution rms error, coverage, and

1    clutter values are examined, and if 2802 they do not indicate a reasonable match between image

2    and stored pattern, the operation is aborted 2803 without attempting to run the high resolution

3    module. If the low resolution module produces a good match, high resolution module is run

4    2804. If the high resolution module does not produce good results 2806, it usually means that the

5    image is out of focus, and the user is so-warned. In some embodiments, when this happens, the

6    low resolution results are used instead of the high resolution results 2808. If the results of the

7    high resolution module are acceptable, the results of the high resolution module are provided to

8    the user 2810 for interpretation, or further processing, according to the particular application.

9    In a preferred embodiment, an overall match score is computed for each resolution step that is

10   equal to the coverage value minus half the clutter value. The low resolution results are used

11   instead of the high resolution results if the high resolution match score is less than some fraction

12   of the low resolution match score. In a preferred embodiment, the fraction used is 0.9.

13   In a preferred embodiment, the methods of co-pending US patent application Serial No.

14   _____, filed 11/14/97, entitled "Efficient, Flexible Digital Filtering", and co-pending US

15   patent application Serial No. 08/727,721, filed 10/07/96, entitled "Fast, Inexpensive, Subpixel

16   Edge Detection" are used for feature extraction, Cognex Corporation's PatQuick™ tool is used to

17   determine the starting pose, and the multi-resolution style of figure 26 is used. The following

18   parameter settings are used for feature extraction by default. Many other strategies can be devised

19   to suit specific applications.

20   For training the low resolution pattern **2600**, and corresponding run-time module **2620**, the image

21   is sub-sampled by sub-sampler **320** by an equal amount in x and y given by the formula

1 $$floor\left(\sqrt{\dfrac{\sqrt{wh}}{8}}\right)$$

2 where $w$ and $h$ are the width and height, respectively, of the pattern 100 in pixels and the *floor*

3 function gives the largest integer that is less than or equal to its argument. Note that sub-sampling

4 by n means taking every $n^{th}$ pixel. The low-pass filter 310 uses a filter size parameter ("s" in co-

5 pending US patent application Serial No. _____, filed 11/14/97, entitled "Efficient,

6 Flexible Digital Filtering") equal to one less than the computed sub-sample amount. The

7 Cartesian to polar conversion module 340 multiplies the gradient magnitude values by 2.0 to

8 improve precision at the low end, where most gradient values lie.

9 For training the high resolution pattern 2610, and corresponding run-time module 2630, the low-

10 pass filter 310 and the sub-sampler 320 are set to pass the source image 300 unmodified.

11 As part of its operation, the PatQuick™ tool reports a "contrast" value in gray levels that is the

12 median gradient magnitude of the pixels in the image on which it is run that correspond to the

13 trained pattern. In a preferred embodiment, this contrast value is used to set the default noise

14 threshold for the peak detector 350. Many other schemes for setting noise thresholds are known

15 in the art that can be used to achieve equivalent results.

16 In said preferred embodiment, PatQuick™ is run on the training image 100 and the contrast value

17 reported by the tool is saved as part of the pattern parameters 220. For training the low resolution

18 pattern 2600, the peak detection module 350 uses a noise threshold equal to 10 gray levels. For

19 training the high resolution pattern 2610, the peak detection module 350 uses a noise threshold

20 equal to one-quarter of said saved contrast.

1 For the run-time image 130, when the PatQuick™ tool is used to determine the starting pose the

2 contrast value it reports is examined. For both the low resolution run-time module 2620, and the

3 high resolution run-time module 2630, the peak detection module 350 uses a noise threshold

4 equal to that used for the corresponding pattern 2600 or 2610, but in each case multiplied by the

5 ratio of run-time contrast to the saved train-time contrast.

6 The preferred embodiments described herein use a six-degree-of-freedom coordinate transform to

7 represent the mapping between points in the image and points in the pattern (i.e. the pose), and a

8 least-squares fitting to determine how to use the information provided by the field to modify a

9 given pose so as to produce a new pose that represents a better correspondence between image

10 and pattern features. Many other arrangements can be devised by those of ordinary skill in the art

11 for achieving similar results within the scope of the invention. These other arrangements may

12 have advantages in certain specific applications.

13 For example, the six degree of freedom coordinate transform can be replaced with other analytic

14 models of the mapping between points in the image and points in the pattern. One useful such

15 model is the well-known perspective transform. Another useful model is one that corrects for

16 lens distortions, such as that produced by so-called "fisheye" lenses. In these cases a different

17 least squares solution would be used, and appropriate changes would be made to the pose

18 element 1500, the compose module 1505, the normal tensor module 1510, the map module 1520,

19 the rotate module 1530, the sums module 1535, and the solve module 1540. The image dipole

20 1515, field module 1525, evaluate module 1545, and field dipole evaluation module 1550 need

21 not change.

1    In other arrangements, the least squares method can be replaced with other well-known methods

2    for fitting data. In such arrangements, appropriate changes might be made to the rotate module

3    1530, the sums module 1535, and the solve module 1540. Alternatively, one or more of these

4    modules might be replaced by different modules that are required for the fitting method to be

5    used.

6    In still other arrangements, a non-analytic mapping between points in the image and points in the

7    pattern, such as a 2-dimensional lookup table with interpolation, may be used. In such an

8    arrangement, the pose 1500 is a lookup table mapping image points to pattern points, and the

9    map module 1520 does the lookup and interpolation. The field module 1525 and evaluate module

10   1545 can be used without modification. The compose module 1505, normal tensor module 1510,

11   rotate module 1530, sums module 1535, and solve module 1540 are not used. Instead, an

12   intermediate lookup table is produced as follows. For every image dipole 1515, an entry is made

13   in the intermediate lookup table by adding the force vector obtained from the field module 1525

14   to the mapped position from map module 1520. Along with this field-corrected position, the

15   weight obtained from the evaluate module 1545 is also stored in the intermediate table entry.

16   The intermediate table thus produced may be sparse, in that many points will not have been filled

17   in, and it may have errors caused by the occasional unreliable image dipole. It can be used,

18   however, to produce a new pose 1500 by applying a *smoothness* constraint. For example, each

19   element of the new pose can be determined by a weighted mean or median of some neighborhood

20   of corresponding elements of the intermediate table. Other methods for using smoothness as a

21   constraint are well-known in the machine vision literature.

1    Other modifications and implementations will occur to those skilled in the art without departing

2    from the spirit and the scope of the invention as claimed. Accordingly, the above description is

3    not intended to limit the invention, except as indicated in the following claims.

4